Dawson College

Laboratory Report

Physics Comprehensive Examination

**Project Title: BlueBot**

Written by:

Myles Elman

Justin Ferreira

Submitted to: Joel Trudeau

Date of the Lab Report Submission: 2025/05/30

**Introduction**

In recent years, mobile robotics has become increasingly popular not just in industry but also in educational and hobby settings. Platforms like Arduino have made it possible for students to design and build functional robots with limited resources. BlueBot was developed as part of this trend, integrating fundamental hardware components with software to create a simple, customizable remote-controlled vehicle. This report outlines the design, development, and testing of BlueBot, a Bluetooth-controlled Arduino robot created during our comprehensive examination (CE) in the engineering physics course with Manuel Toharia Zapata. As future mechanical and electrical engineering students, this project offered a valuable opportunity to deepen our understanding of electronics and automation through hands-on experience. We would see this as an opportunity to utilise our knowledge gained in our future schooling years.

**Project Description**

**Objective:** The primary objective of BlueBot was to build a functional two-wheel robot that could be remotely controlled via a smartphone/computer application using Bluetooth. The goal extended beyond basic movement because it included gaining a working understanding of hardware-software integration, efficient motor control, and user-directed wireless navigation.

**Inspiration:** We were inspired by online Arduino projects and tutorials that showcased Bluetooth-controlled robots. For instance, one particular project that stood out was a YouTube tutorial titled "How to make a Bluetooth control car with a 2WD smart car kit" by SriTu Hobby. This tutorial demonstrated a similar two-wheel drive robot controlled via a mobile app using the HC-05 module. Seeing the clear explanation and results helped us believe that we could build a version tailored to our needs. We also saw another project in a previous McGill Physics Hackathon that had a car with LED lights incorporated and we thought that this was really cool. We have always had a large passion for building things and LEGOs and wanted to fabricate something cool.

**Design and Planning:** Initially, we were a bit behind in starting this project, as we had a lot of other examinations and work in other classes. However, our planning began soon after and was in full force. Initially, we planned for a power-separated system using USB power banks. However, after testing, the final design was simplified for reliability and ease of use. Together, we sketched rough wiring diagrams and physically laid out the components on the chassis to optimize space and minimize wiring issues. We also had to consider current demands and voltage compatibility to avoid damaging the components, so we needed to get a battery with a large enough voltage to sustain all of this. During our initial team meeting, we discussed how the commands from the Bluetooth module would be interpreted and translated into motor actions. We changed our design many times as we worked through this project, as numerous things did not work and we sometimes found simpler ways of doing things. We watched many YouTube videos for assistance and also got help from other students. We primarily spent our time working in the physics tutorial room 7A.2.

**Implementation**

**Hardware Setup:** The components used are as indicated: one Arduino uno compatible board, an L298N Dual H-Bridge Motor Driver, two 6V DC geared motors with rubber wheels, an HC-05 Bluetooth module, an MDF laser-cut robot chassis, male-to-female jumper wires and a 9V battery. As mentioned before, the system was initially powered using two rechargeable USB power banks. One powered the Arduino board via the USB port, while the other supplied current to the motor driver. We also had resistors and a breadboard (Diagram 1). This dual-power strategy resulted in more stable performance during extended operation, but it was too complicated and we realized there was an easier way. Thus, in our final design, we simplified the system by removing the breadboard, eliminating the second Arduino, and powering the entire circuit with a single 9V battery connected directly to the main Arduino board. The HC-05 module and L298N driver were wired using standard jumper wires without any resistors or voltage divider (Diagram 2). This setup reduced wiring complexity, improved reliability, and allowed for a more compact and manageable robot.

Diagram 1: Initial Configuration

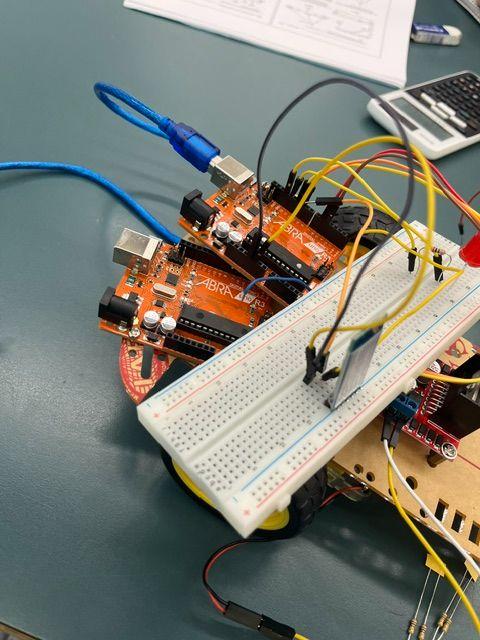
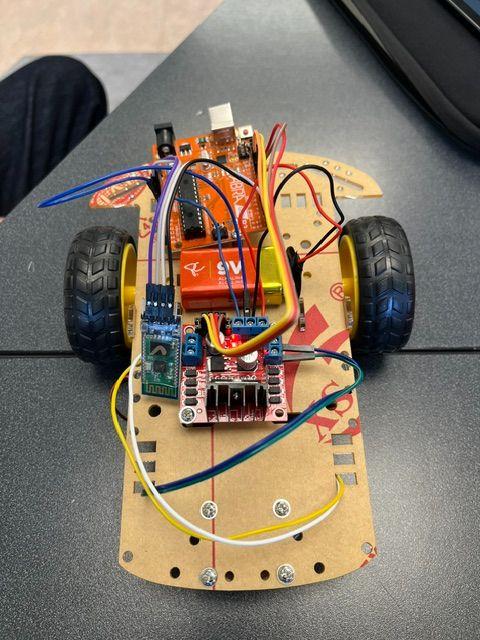
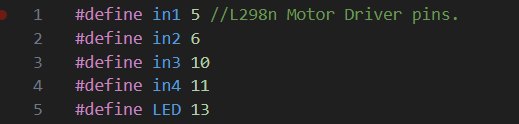
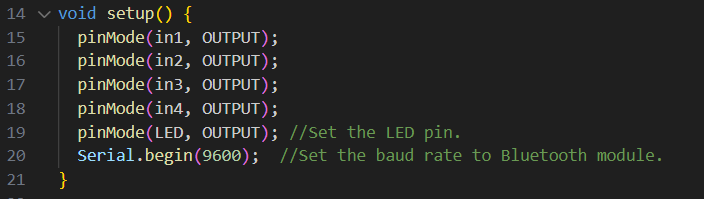


Diagram 2: Final Configuration

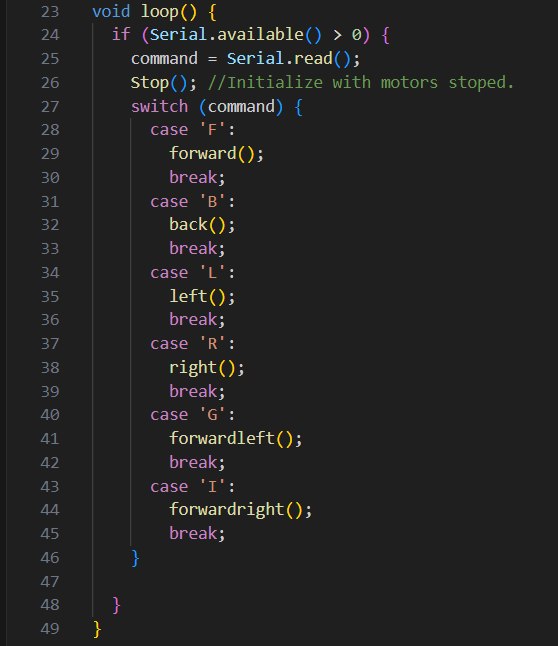


**Software Development:** The software was written using the Arduino IDE using basic C++ functions. No external libraries were needed, which made the code lightweight and easy to debug. The core functionality relies on receiving commands over Bluetooth and mapping those commands to specific motor actions using PWM signals. To begin, we defined the motor control pins connected to the L298N motor driver. These pins were set as outputs in the setup function:

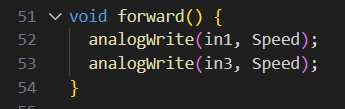




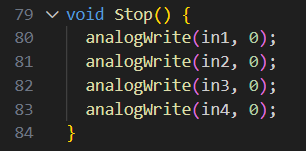
The robot continuously checks for incoming Bluetooth commands within the loop function. When a character is received, such as 'F' for forward or 'L' for left, the corresponding function is called after first stopping the motors to prevent any overlap or unintended motion:



Each direction function controls the motors by adjusting the PWM signal using analogWrite(). For example, the forward function applies speed to IN1 and IN3:



We initially set the speed to 220 but found it too fast during testing. To improve control and stability, we reduced the value to 150. We also implemented a Stop() function to ensure all motors shut off between commands.



To send the commands, we used the “Arduino Bluetooth Controller” app on an Android tablet. The Car Controller interface allowed us to send single-letter commands that the Arduino could interpret via its serial input. Although the HC-05 module did not work with iOS, it was compatible with this Android-based system.

**Sensor Documentation:** In this version of BlueBot, no sensors were integrated. However, we planned to add an HC-SR04 ultrasonic sensor in future revisions. This sensor emits ultrasonic pulses and measures the time it takes for the echo to return, allowing it to calculate distance to nearby objects. It would be connected to a digital pin for triggering the pulse and another pin to read the echo time.

**Results**

BlueBot successfully completed all core functionalities required by the project: it moved forward, backward, and turned left and right upon receiving commands via Bluetooth. These commands were transmitted using a Bluetooth application titled “Arduino Bluetooth Controller” and each directional command (F, B, L, R,) triggered the appropriate motor responses.

The robot was tested on both smooth and slightly uneven surfaces. On smooth indoor flooring (our house), movement was consistent and direction changes were sharp. On less even terrain (library carpet), some minor drift was observed, but overall control remained reliable. We tested runtime and it revealed that the single 9V battery could support continuous operation for over 30 minutes before slowdown occurred. Also, our wiring connections remained secure throughout testing thanks to the straightforward layout using male-to-female jumper wires. Visual indicators, such as the blinking LEDs on the HC-05, also helped confirm that the system was correctly paired and responsive throughout trials.

**Analysis**

The results demonstrated that BlueBot achieved its core objectives: reliable wireless control, consistent movement, and stable electrical performance. On smooth surfaces, like hardwood flooring, BlueBot maintained straight trajectories and turned sharply. However, on carpeted surfaces, we noticed a slight drift, especially veering to the left, which suggests a minor imbalance in motor torque or wheel alignment. This highlighted the importance of mechanical symmetry and could be addressed in future iterations by incorporating encoders or PID control for directional correction. Also, the robot was initially too fast and we altered its max speed from 220 to 150. This is because BlueBot is very light and can accelerate too quickly. Another key takeaway was the power system’s efficiency. A single 9V battery sustained over 30 minutes of operation without major voltage drops. In future versions, switching to an HC-10 module or BLE-compatible chip would allow broader device compatibility without the need for reconfiguration. Overall, the project reinforced the value of iterative design, careful power planning, and user-focused component selection — all critical concepts in embedded systems and robotics engineering.

**Discussion and Conclusion**

Reflecting on this project, we are proud of how much we were able to learn and accomplish in a relatively short amount of time. One of us started with a basic knowledge of Arduino and left with a much more confident understanding of electronics, circuit design, and C++ programming for embedded systems. One of the biggest takeaways was how critical it is to test incrementally and isolate issues. Many challenges, like unresponsive motors or weird Bluetooth behavior, were solved by breaking the problem down into smaller parts. For instance, one specific issue we encountered was with the Bluetooth module itself. The HC-05 module is not compatible with iPhones, and we both have iPhones. As a result, we could not control the robot using our phones as originally planned. To work around this, we first connected the HC-05 to a Windows laptop and used the serial monitor to direct movement. However, this also raised a compatibility issue with Microsoft and we had to change designs. Thus, we monitored the HC-05’s code and changed its program to be compatible with most devices. This unexpected problem taught us the importance of checking hardware compatibility across devices during the planning phase. We also faced challenges with wiring stability. Loose jumper wires during early tests led to random disconnections and motor stalling. We resolved this by tightening connections and using shorter wires where possible. Initially, we used two Arduinos and USB power banks to separate logic and motor power, but this caused instability. Simplifying the build to one Arduino and a 9V battery improved performance. Among the most rewarding moments was seeing the robot move for the first time in response to our commands, which made the hours of debugging and wiring feel worth it. We also gained more appreciation for the physical side of robotics, like securing components properly, managing wires, and planning layout to avoid interference. It was our first big project and we had a lot of fun assembling the robot and seeing everything come together. We also learned a lot about github and how to showcase our works, which we will definitely use in the near future. Although we faced some setbacks, we learned how to troubleshoot together. As mentioned earlier, in the future, we would like to expand BlueBot with autonomous features such as obstacle detection using the HC-SR04 sensor, as well as line-following capabilities. We could switch to an HC-10 bluetooth module so BlueBot is compatible with all devices. We would also like to make the robot prettier, like adding LED lights and surrounding it with LEGOs. Overall, this project strengthened our interest in robotics, and we are truly excited to continue exploring more complex systems in university and beyond.

**References**

* “Arduino 4WD Robotic Car Assemble Guide - STEManGo Electronics Version.” *YouTube*, uploaded by STEManGo Electronics, 30 August 2018, <https://youtube.com/playlist?list=PLBwTxof3FzUULl9hXM7B3GuvG1qwXYoSD&si=9tOxaIm-vDCDA2mj>
* “How to make a Bluetooth control car with a 2WD smart car kit | 2WD Bluetooth car Arduino.” *YouTube*, uploaded by SriTu Hobby, 7 November 2024, <https://youtu.be/z524YqJvyak?si=r1xDZb6Oqwg-XVWT>
* Dawson Virtual Lab, *Dawson College*, <https://englab.dawsoncollege.qc.ca/>

**Acknowledgements**

We would like to thank our supervisor, Joel Trudeau, for always being available when we had questions, especially during wiring and testing. This project wouldn’t have been possible without his support and encouragement. He helped us learn a lot of new things as we built this robot that we will definitely take into the future, especially as we pursue engineering careers in university. We would also like to thank the Dawson lab technicians located on the seventh floor for always being available and willing to let us borrow materials such as screwdrivers and wires. Every time we worked on our project and we needed a random material or tool, we could always count on them to be available and ready to assist.